IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE

IEEE

Membership Publications/Services Standards Conferences Careers/Jobs

IEEE Xplore®
RELEASE 1.5

Welcome
United States Patent and Trademark Office

Help FAQ Terms IEEE Quick Links » Search Results
Peer Review

Welcome to IEEE Xplore®
- Home
- What Can I Access?
- Log-out

Tables of Contents
- Journals & Magazines
- Conference Proceedings
- Standards

Search
- By Author
- Basic
- Advanced

Member Services
- Join IEEE
- Establish IEEE Web Account
- Access the IEEE Member Digital Library

Print Format

Your search matched **[0]** of **[957328]** documents.

You may refine your search by editing the current search expression or entering
a new one the text box. Then click search Again.

(abdallah <in> au)and (al-dajani <in> au)

Search Again

OR

Use your browser's back button to return to your original search page.

Results:

No documents matched your query.

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search
Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting
No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top

WEST

Generate Collection | Print

L2: Entry 3 of 5 File: USPT Jan 2, 2001

US-PAT-NO: 6170052
DOCUMENT-IDENTIFIER: US 6170052 B1

TITLE: Method and apparatus for implementing predicated sequences in a processor with renaming

DATE-ISSUED: January 2, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Morrison; Michael J.	Santa Clara	CA		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Intel Corporation	Santa Clara	CA			02

APPL-NO: 09/ 002016 [PALM]
DATE FILED: December 31, 1997

INT-CL: [07] G06 F 21/06

US-CL-ISSUED: 712/236; 712/23, 712/226, 712/240
US-CL-CURRENT: 712/236; 712/226, 712/23, 712/240

FIELD-OF-SEARCH: 712/23, 712/29, 712/26, 712/209, 712/211, 712/214, 712/240, 712/207, 712/236, 712/239, 712/226

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected | Search ALL

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
5452426	September 1995	Papaworth et al.	395/375
5721855	February 1998	Hilton et al.	395/394
5740393	April 1998	Vidwans et al.	395/391
5889982	March 1999	Rodgers et al.	395/570

ART-UNIT: 273

PRIMARY-EXAMINER: Follansbee; John A.

ASSISTANT-EXAMINER: Nguyen; Dzung C.

ATTY-AGENT-FIRM: Blakely, Sokoloff, Taylor & Zafman LLP

ABSTRACT:

Systems, apparatus, and methods are disclosed for generating pairs of conditional instructions corresponding to special predicate sequences from single instructions having a predicate. These pairs of conditional instructions update a destination register regardless of the truth or falsity of the predicate. The destination register is renamed to a new physical location. In this manner, register renaming can be used with predicate sequences to gain performance efficiencies and to overcome limitations of the prior attempted approaches.

34 Claims, 9 Drawing figures

WEST

Generate Collection Print

L2: Entry 3 of 5 File: USPT Jan 2, 2001

DOCUMENT-IDENTIFIER: US 6170052 B1
TITLE: Method and apparatus for implementing predicated sequences in a processor with renaming

Brief Summary Text (17):
In one aspect of the present invention, a method is provided for renaming registers. The method includes renaming a destination register of a first conditional micro-op with a physical register, and renaming a destination register of a second conditional micro-op with the physical register. The first conditional micro-op belongs to a special predicated sequence. The second conditional micro-op belongs to the same special predicated sequence.

Brief Summary Text (18):
In another aspect of the present invention, an apparatus is provided. The apparatus includes a renamer to rename the destination registers of two conditional micro-ops with a single physical register in response to the two conditional micro-ops having the same destination register and belonging to a single special predicated sequence. The apparatus includes a rename table coupled to the renamer. The renamer is adapted to write the identity of the physical register that renames each destination register to a location of the rename table. The apparatus includes at least one execution unit coupled to the renamer. The execution unit is adapted to execute a portion of the micro-ops from the renamer.

Drawing Description Text (9):
FIG. 7 shows machine instructions used by a first embodiment of the processor of FIG. 5 to identify the second conditional micro-op of a special predicated sequence;

Drawing Description Text (11):
FIG. 9 is a flowchart for a second embodiment for identifying conditional micro-ops of special predicated sequences in the method of FIG. 6.

Detailed Description Text (4):
The specific embodiments include hardware structures adapted to use special predicated sequences. The special predicated sequences include paired conditional instructions that update a destination register regardless of the truth or falsity of the predicate. The hardware structures insure that logical destination registers of the paired conditional instructions from the same special predicated sequence are renamed with the same physical register. Therefore, dependent instructions have a source register that is a destination address of a pair of conditional micro-ops of the special predicated sequence.

Detailed Description Text (5):
FIG. 3 illustrates a class of special predicated sequences 30 that is employed by some specific embodiments. The special predicated sequence 30 includes a predicate evaluating micro-op 31, which stores a value of logic 1 and of logic 0 in a predicate register P.sub.x when the predicate is true and is false, respectively. The special predicated sequence 30 also includes first and second conditional micro-ops 32, 33. The conditional micro-ops 32, 33 are executed when the condition P.sub.x is true and false, respectively. The first and second conditional micro-ops 32, 33 have the same logical destination register Z. The second conditional micro-op 33 restores a data previously stored in the register Z back in the same register Z when the predicate is false. Thus, the special predicated sequence 30 updates the contents of the register Z for both values, i.e. true or false, of the predicate P.sub.x. A micro-op 34 that is

both later in the instruction sequence and contains Z as a source address is dependent on both of the conditional micro-ops 32, 33 of the special predicated sequence 30.

Detailed Description Text (6):
In the specific embodiments, conditional micro-ops 32, 33 of a special predicated sequence 30 are paired so that destination addresses of the conditional micro-ops are updated whether or not the predicate is true. The specific embodiments may employ different types of conditional micro-ops and different numbers of conditional micro-ops than the micro-ops 32, 33 of FIG. 3, but the destination registers of the conditional micro-ops of a special predicated sequence are updated whether the predicate is true or false. In the specific embodiments described-below, the conditional micro-op 33 that restores a data word of the register Z in the same register Z is the third micro-op in the special predicated sequence 30. The present invention is intended to also include embodiments wherein the conditional micro-op of the special predicated sequence that restores a data word in the register in the same register is at another position in the special predicated sequence.

Detailed Description Text (7):
FIG. 4 illustrates one embodiment for renaming the special predicated sequence 30 of FIG. 3. In a renamed special predicated sequence 36, a physical destination register Z.sub.1 replaces the logical destination register Z of the original special predicated sequence. In the renamed special predicated sequence 36, the destination address of both renamed conditional micro-ops 38, 39 is the same physical register Z.sub.1. The source address of the second micro-op 39, which renames the conditional micro-op 33 of FIG. 3, points to a physical destination register Z.sub.2 of a micro-op 40 that renames the micro-op 35 of FIG. 3. The micro-op 35 is the last micro-op of the instruction sequence that both has register Z as a logical destination address and is earlier in the instruction sequence than the predicated sequence 30. Thus, the logical source register of the second conditional micro-op 33 is renamed to point to the destination register of the last micro-op preceding the special predicated sequence 36. A dependent micro-op 34 with a source address that is a destination address of the conditional pair 32, 33 of FIG. 3 retains a source address that points to the renamed destination address of the two renamed conditional micro-ops 38, 39 of the special predicated sequence 36 of FIG. 4. The renamed special predicated sequence 36 updates the destination register Z.sub.1, whether the predicate 37 is true or is false and a dependent micro-op 41 points to the same physical renamed destination register.

Detailed Description Text (8):
FIG. 5 illustrates a portion of an out-of-order processor 50. A fetcher 52 sequentially retrieves macro-instructions from memory and/or caches without waiting for previously fetched instructions to be executed. The fetcher 52 sends the macro instructions to one or more decoders 54, 56. The decoders 54, 56 translate the macro-instructions into executable micro-ops. The decoders 54, 56 translate one or more types of conditional instruction into the special predicated sequence, including the predicate evaluating micro-op 31 and the associated pair of conditional micro-ops 32, 33, as illustrated in FIG. 3. The decoders 54, 56 send the micro-ops to a renamer 58.

Detailed Description Text (9):
The renamer 58 employs blind renaming to replace logical registers appearing as destination addresses in micro-ops with different physical registers, i.e. removing artificial write-after-write and write-after-read dependencies. The renamer 58 recognizes paired conditional micro-ops associated with special predicated sequences and performs register renaming in a manner that is consistent with the method illustrated in FIG. 3. The renamer 58 records assignments of physical registers for logical destination registers in a rename table 60.

Detailed Description Text (11):
FIG. 6 is a flowchart illustrating the steps of a method 66 for executing conditional instructions with special predicated sequences on the out-of-order processor 50 of FIG. 5. At block 68, the method begins by decoding a conditional instruction into a predicate evaluating micro-op 31 and an associated pair of conditional micro-ops 32, 33 having the form illustrated in FIG. 3. At block 70, the predicate and pair of conditional micro-ops are sequentially sent from one of the decoders 54, 56 to the renamer 58. At block 72, the logical destination register of the first conditional

micro-op 32 of the predicate 30 is renamed with a physical register. At block 74, a micro-op received at the renamer 58 is identified as the second conditional micro-op 33 associated with the same predicated sequence 30. At block 76, the physical register that renames the logical destination register of the first conditional micro-op 32 is assigned to be the physical register that renames the logical destination register of the second conditional micro-op 33 of the pair 32, 33. At block 78, the logical register for the source address of the second conditional micro-op 33 is replaced with a physical register that renames the same logical register in a destination address of the last micro-op 40 in the instruction sequence that is earlier than the predicated sequence 30. At block 80, the predicate evaluating micro-op 37 and the associated pair of renamed conditional micro-ops 38, 39 are executed on one or more of the execution units 62, 64.

Detailed Description Text (12):
Referring to FIG. 7, the renamer 58 of one embodiment uses one bit of the binary machine word 82 of a micro-op to determine whether the micro-op is the second conditional micro-op 33 of a pair associated with a special predicated sequence 30. The binary machine word 82 has a number of binary bits 84 for identifying the form of the micro-op. At least one bit is a condition bit 86 that identifies the micro-op as the conditional micro-op 33 of a pair associated with a special predicated sequence 30. For micro-ops that are not the second conditional micro-op 33 of a special predicated sequence 30, the machine word 88 for the micro-op has the value logic 1 in its condition bit 90. For a micro-op that is the second conditional micro-op 33 of a predicated sequence 30, a machine word 92 for the micro-op has the value logic 0 in its condition bit 94. In this embodiment, the decoders 54, 56 initialize the condition bit 86 to have a value of logic 1 or logic 0 so that the renamer 58 can identify second conditional micro-ops 33 of special predicated sequences 30.

Detailed Description Text (13):
In other embodiments, the machine code for micro-ops does not have a condition bit 86 for identifying the second conditional micro-op 33 of the predicated sequence 30. Instead, the renamer 58 has hardware structures (not shown) for identifying the second conditional micro-op 33 of the predicated sequence 30. The hardware structures read more than one bit of the machine code for micro-ops. In one specific embodiment, the hardware structures trigger when the execution of a first micro-op depends on a logic value stored in a predicate register, e.g., the register P.sub.x. Then, the renamer 58 looks for a later micro-op that depends on the value stored in the register P.sub.y conjugated to P.sub.x and has the same destination address. The renamer 58 identifies such a micro-op as the second conditional micro-op 33 of the same predicated sequence 30. The dependence on the conjugated predicate registers P.sub.y and P.sub.x can be implemented with neighboring bits of the machine words for micro-ops. The present invention is intended to cover embodiments employing other methods and apparatus for identifying second conditional micro-ops that an ordinary person in the art would be able to construct and use, without undue experimentation, in light of the present disclosure.

Detailed Description Text (14):
FIG. 8 illustrates a method 108 for using the condition bit 86 of the machine word 82 for a micro-op to identify the second conditional micro-op 33 of a special predicated sequence 30. At block 110, the machine word 82 for a micro-op from one of the decoders 54, 56 is sent to the renamer 58. At block 112, the condition bit 86 of the machine word 82 for the micro-op is read with hardware of the renamer 58. At block 114, whether the micro-op is a second conditional instruction 33 of a special predicated sequence 30 is determined from the value 90, 94, i.e. logic 1 or logic 0, of the condition bit 86. At block 116, the logical destination register of the micro-op is replaced with an unassigned physical register is replaced in response to the micro-op not being a second conditional micro-op 32 of a special predicated sequence 30. At block 118, a physical register assigned to the logical destination register of the micro-op is looked up in the rename table 60, and assigned the same physical register is assigned for the logical destination register of the micro-op in response to determining, at block 114, that the micro-op is a second conditional micro-op 33 of a special predicated sequence 30. At block 118, the physical register in the rename table 60 is the destination register of the first conditional micro-op 32 of the same special predicated sequence 30. At block 120, the logical source register of the micro-op in the renamer 58 is replaced with the physical register that was previously

assigned to the logical register. The physical register previously assigned to the logical source register renames the destination address of the latest micro-op 35 in the instruction sequence that both has the logical register as a destination address and is earlier than the special predicated sequence 30 in the instruction order.

Detailed Description Text (15):
FIG. 9 illustrates a second method 122 for identifying the second conditional micro-op 33 of a special predicated sequence 30. At block 124, the machine word 82 for a micro-op is sent from one of the decoders 54, 56 to the renamer 58. At block 126, the renamer 58 determines whether the execution of the micro-op is dependent on the value stored in a predicate register. If the execution depends the value in a predicate register, the micro-op is a conditional micro-op 32, 33 of a special predicated sequence 30. The machine word for the micro-op may have several bits for indicating dependencies on predicate registers. At block 128, if the micro-op is a conditional micro-op 32, 33 of a special predicated sequence 30, the renamer 58 determines whether the predicate register of the micro-op is the conjugate of a predicate register appearing in an earlier micro-op in the instruction sequence. At block 130, the renamer 58 determines whether the present micro-op and the earlier micro-op have the same destination register. If the two micro-ops have the same destination register, the present micro-op is determined to be the second conditional micro-op 33 of the same special predicate sequence 30 at block 132. At block 134, the renamer 134 determines that the micro-op is a first conditional micro-op 32 if the micro-op is conditional on the value stored in a predicate register but not a second conditional micro-op 33.

CLAIMS:

1. A method of renaming comprising:

generating a special predicate sequence for a micro-op instruction having a predicate, the special predicate sequence including a first and second conditional micro-op instructions for updating a destination register of the micro-op instruction regardless of the truth or falsity of the predicate;

renaming the destination register of the first conditional micro-op instruction with a physical register; and

renaming the destination register of the second conditional micro-op instruction with the physical register.

4. The method of claim 2, further comprising sending the micro-ops of the special predicated sequence to a renamer in consecutive cycles.

7. The method of claim 1, further comprising determining whether the second micro-op is both a conditional micro-op of a special predicated sequence and is for restoring a data word to a destination register that previously stored the data word, the step of renaming a destination register of a second conditional micro-op being performed in response to determining that the second micro-op is both a conditional micro-op of a special predicated sequence and is for restoring a data word to a destination register that previously stored the data word.

11. The method of claim 1, wherein the steps of renaming include renaming micro-ops of a first type of predicated sequence, the first type of predicated sequence being a sequence of micro-ops that consists of a predicate evaluating micro-op, a third micro-op, and a fourth micro-op, the third micro-op being executed in response to the predicate being true, and the fourth micro-op being executed in response to the predicate being false, the fourth micro-op being for restoring a data word to a register that previously stored the data word.

14. An apparatus, comprising:

a special predicate sequence generator to generate a single special predicated sequence from a micro-op instruction having a predicate, the special predicate sequence including at least two conditional micro-ops for updating a same destination register regardless of the truth or falsity of the predicate;

a renamer to rename the destination register of the two conditional micro-ops with a single physical register;

a rename table coupled to the renamer, the renamer being adapted to write the identity of the physical register that renames each destination register to a location of the rename table; and

at least one execution unit coupled to the renamer, the execution unit being adapted to execute a portion of the micro-ops from the renamer.

15. The apparatus of claim 14, further comprising at least one decoder having an output terminal coupled to an input terminal of the renamer, the decoder to produce the special predicated sequence of micro-ops in response to receiving a special sequence of at least one instruction.

17. The apparatus of claim 16, wherein the renamer is capable of determining that a particular micro-op is a conditional micro-op of a special predicated sequence in response to determining that the condition bit of the machine word for the particular micro-op has a preselected value, the particular micro-op for restoring a data word to a register that previously stored the data word.

18. The apparatus of claim 15, wherein the renamer is adapted to receive the micro-ops of each special predicated sequence in consecutive cycles and the decoder is adapted to send the micro-ops to an input terminal of the renamer in consecutive cycles.

21. The apparatus of claim 14, wherein the renamer is adapted to store the identity of a physical register that previously renamed a logical register to a special location in response to receiving the first conditional micro-op of the single special predicated sequence, the destination register of the first conditional micro-op being the logical register.

30. An out-of-order processor, comprising:

a fetcher;

at least one decoder to receive instructions from the fetcher, the decoder being capable of decoding a sequence of at least one instruction into a special predicated sequence, the special predicate sequence corresponding to one of the at least one instruction and including a first and second conditional micro-op instructions for updating a destination logical register of the instruction regardless of the truth or falsity of a predicate of the instruction; and

a renamer being adapted to rename the first and second conditional micro-ops of the special predicated sequence and being coupled to an output terminal of the decoder, the renamer to replace the destination logical register appearing as a destination address in the first and second conditional micro-op instructions of the special predicated sequence with a single physical register.

WEST Search History

DATE: Thursday, July 31, 2003

Set Name side by side	Query	Hit Count	Set Name result set
	DB=JPAB,EPAB,DWPI,TDBD; PLUR=NO; OP=ADJ		
L4	predicat$3 and L3	7	L4
L3	microinstruction$1 or micro-instruction$1 or microcode or micro-code or micro-operation$1 or microoperation$1 or micro-op$1 or microop$1 or micro-program$1 or microprogram$1	11101	L3
	DB=USPT,PGPB; PLUR=NO; OP=ADJ		
L2	l1 with predicat$3	5	L2
L1	microinstruction$1 or micro-instruction$1 or microcode or micro-code or micro-operation$1 or microoperation$1 or micro-op$1 or microop$1 or micro-program$1 or microprogram$1	11999	L1

END OF SEARCH HISTORY

PORTAL

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ○ The Guide ◉ The ACM Digital Library

(+microcode or +microinstruction or +microoperation or +microop) +predic

Feedback R

Terms used microcode or microinstruction or microoperation or microop predicat

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

☐ Open results in a new window

Try a

Try th

Results 1 - 1 of 1

1 Some Aspects of High-Level Microprogramming
Subrata Dasgupta
September 1980 ACM Computing Surveys (CSUR), Volume 12 Issue 3
Full text available: pdf(2.32 MB) Additional Information: full citation, references, citings, index terms

Results 1 - 1 of 1

Your search matched **2** of **957328** documents.

A maximum of **2** results are displayed, **15** to a page, sorted by **Relevance** in **descending** order.
You may refine your search by editing the current search expression or entering a new one the text box.

Then click **Search Again.**

(microcode or microinstruction or microoperation or microop) and predicat*

Search Again

Results:
Journal or Magazine = **JNL** Conference = **CNF** Standard = **STD**

1 **An automata-theoretic approach to local microcode generation**
Langevin, M.; Cerny, E.;
Design Automation, 1993, with the European Event in ASIC Design. Proceedings. [4th] European Conference on , 22-25 Feb. 1993
Page(s): 94 -98

[Abstract] [PDF Full-Text (404 KB)] **IEEE CNF**

2 **An extended OBDD representation for extended FSMs**
Langevin, M.; Cerny, E.;
European Design and Test Conference, 1994. EDAC, The European Conference on Design Automation. ETC European Test Conference. EUROASIC, The European Event in ASIC Design, Proceedings. , 28 Feb.-3 March 1994
Page(s): 208 -213

[Abstract] [PDF Full-Text (484 KB)] **IEEE CNF**

Welcome
United States Patent and Trademark Office

Help FAQ Terms IEEE Quick Links
Peer Review
» Search Results

Welcome to IEEE *Xplore*®

- Home
- What Can I Access?
- Log-out

Tables of Contents

- Journals & Magazines
- Conference Proceedings
- Standards

Search

- By Author
- Basic
- Advanced

Member Services

- Join IEEE
- Establish IEEE Web Account
- Access the IEEE Member Digital Library

Print Format

Your search matched **[0]** of **[957328]** documents.

You may refine your search by editing the current search expression or entering
a new one the text box. Then click search Again.

(abdallah <in> au)and (predicate)

Search Again

OR

Use your browser's back button to return to your original search page.

Results:

No documents matched your query.

IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE

IEEE

Membership Publications/Services Standards Conferences Careers/Jobs

IEEE Xplore®
RELEASE 1.5

Welcome
United States Patent and Trademark Office

Help FAQ Terms IEEE Quick Links » Search Results
Peer Review

Welcome to IEEE Xplore®
- Home
- What Can I Access?
- Log-out

Tables of Contents
- Journals & Magazines
- Conference Proceedings
- Standards

Search
- By Author
- Basic
- Advanced

Member Services
- Join IEEE
- Establish IEEE Web Account
- Access the IEEE Member Digital Library

Print Format

Your search matched **[0]** of **[955444]** documents.

You may refine your search by editing the current search expression or entering
a new one the text box. Then click search Again.

(al-dajni <in> au)and (predicate)

Search Again

OR

Use your browser's back button to return to your original search page.

Results:

No documents matched your query.

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search
Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting
No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top

IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE

IEEE

Membership Publications/Services Standards Conferences Careers/Jobs

IEEE Xplore®
RELEASE 1.5

Welcome
United States Patent and Trademark Office

Help FAQ Terms IEEE Quick Links

» Search Results

Peer Review

Welcome to IEEE Xplore®
- Home
- What Can I Access?
- Log-out

Tables of Contents
- Journals & Magazines
- Conference Proceedings
- Standards

Search
- By Author
- Basic
- Advanced

Member Services
- Join IEEE
- Establish IEEE Web Account
- Access the IEEE Member Digital Library

Print Format

Your search matched **[0]** of **[957328]** documents.

You may refine your search by editing the current search expression or entering
a new one the text box. Then click search Again.

(abdallah <in> au)and (al-dajani <in> au)

Search Again

OR

Use your browser's back button to return to your original search page.

Results:

No documents matched your query.

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search
Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting
No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top